# Chinese electronic health record analysis for recommending medical treatment solutions with NLP and unsupervised learning

*Junmei* Zhong[1,*], and *Xiu* Yi[2]

[1]Inspur USA Inc, Bellevue, WA, USA
[2]Inspur Inc, Jinan, Shandong, China

**Abstract.** Electronic health record (EHR) analysis has become increasingly important in improving the quality of human healthcare. To leverage the full insights from the big EHRs, it is very important to define some application scenarios for which the relevant data can be extracted for training machine learning models to accomplish the expected goals. In this paper, we develop a system on how to recommend medical treatment solutions for patients living in the countryside and small cities when they happen to have schizophrenia but the doctors in the local hospitals do not have sufficient expertise to deal with such challenges. In the EHRs, we take the patients' symptom descriptions as documents and then develop NLP and unsupervised machine learning techniques to analyze such documents to find the relevant and effective treatment solutions provided by medical experts. Extensive experimental results with different vector representations for documents show that the binary keyword vector representation works best to find relevant and effective medical treatment plans and solutions from the EHRs for any input symptom description.

## 1 Introduction

In China, the medical resources are distributed very unevenly. Most of the excellent medical resources are concentrated on the big hospitals in big cities, whereas the countryside and small cities are allocated with a small portion of the medical resources. Furthermore, the quality is far from being satisfactory, but most of the country's population is living there. As a result, it is often very difficult for patients not living in big cities to receive the in-time effective medical treatments when they happen to have some schizophrenia due to the lack of high-quality medical resources. The emerging cloud computing and artificial intelligence (AI) technologies for big data analysis have made it possible to some extent for patients in the countryside and small cities to share some excellent medical resources in the big hospitals. The successful development of such technologies can not only mitigate the lack of medical resources in the whole country but also helps save patients' lives, reduces the worries of patients' families to some extent, and reduces patients' costs by leveraging the insights from big EHRs.

[*] Corresponding author: Zhong.junmei@gmail.com

In this paper, we develop NLP and machine learning techniques to analyze patients' EHRs in our cloud to find the available relevant and effective treatment solutions provided by medical experts in big hospitals according to the input patient's symptom description. The recommender system works in two steps: first, it finds the relevant symptoms from the EHRs according to the input patient's symptom, then it finds their corresponding medical treatment solutions provided by the medical experts. As a result, the symptom matching process is used as an indexing procedure to find the relevant medical treatment solutions. For machine learning, it generally has two typical forms, supervised and unsupervised learning. However, due to the dynamic and random characteristics of patients' symptoms, it is impossible to use supervised learning to accomplish the recommender system, and unsupervised learning is practically feasible for finding the relevant documents in the EHRs with respect to the input symptom. In this paper, each patient's symptom is taken to be a document. For documents' vector representation, we have tried the following representations: Vector space models with the TF-IDF weighting method for tokens, binary representation for tokens, binary representation for keywords, the averaged distributed word embeddings with word2Vec, and the doc2Vec for document representation. For document clustering analysis, we choose the Cosine similarity as the quantitative metric to cluster the relevant documents with respect to any input document: a patient's symptom description. With this recommender system, when a patient in the countryside has schizophrenia and the local hospital doctors do not have sufficient expertise to deal with such challenging cases, the doctors can resort to finding the available effective treatment solutions of the similar symptoms from the EHRs. Furthermore, if necessary, they can find the corresponding doctors' contact information for immediate consultation under urgent situations. The successful development of this system can also benefit doctors in big hospitals when they have patients who happen to have schizophrenia, offering the potential of all weather sharing of the medical resources in the whole country.

## 2 Methodologies

The framework of this recommendation system consists of 3 components: data collection, feature extraction with NLP algorithms for document representation, and the quantitative similarity measurement calculation for clustering relevant documents with respect to any input patient's symptom document.

## 3 Data collection

The EHRs are collected from some big hospitals in Shandong Provinces, China, which are composed of 85 selected tables ranging from outpatient records, inpatient records, lab testing results, patient demographical information, doctor information, hospital information, medication information, diagnosis records, and treatment information. The EHRs are characterized by heterogeneity, high dimensionality, missing values, noise, sparseness, incompleteness, inconsistencies (different hospitals and doctors have different standards and different terms to describe the same diseases), random errors and bias. We first analyse these tables according to the underlying clinic process to construct individual complete clinic records from these 85 tables then we extract each patient's individual symptoms and the corresponding treatment solutions, prescriptions provided by the doctors, together with the doctor's contact information. Afterwards, we take each patient's symptom description in each clinic record as a document for the follow-up analysis with NLP and machine learning methodologies. We totally have 1.01 million such constructed clinic records/documents in our EHRs, and after data cleaning to remove those records without having symptom descriptions and/or medical treatment solutions, we get 300K meaningful records.

## 4 Feature extraction with NLP algorithms

To represent each document with a feature vector for measuring the similarity between different documents, we tokenize the documents to get tokens, and extract the keywords, respectively with Han LP [1], an open source software for Chinese texts tokenization, and then we generate a vector for each document. We have tried different representations, which include the TF-IDF weighting method for tokens, the binary representation for tokens, the binary representation for keywords, the averaged word vector of the word2Vec embeddings [2] for each document, and the vector of doc2Vec [3].

### 4.1 The bag of words (BOW) method

The BOW method makes use of individual words and/or N-Grams as features for document representation, which is called a feature vector in machine learning and pattern recognition. All the tokens and/or N-Grams constitute the vocabulary of the documents. For individual features, we have both binary representation and TF-IDF weighting method to get their values. For the binary representation, it does not count the number of occurrences of the tokens in the documents but only considers the presence and absence of the individual tokens. If a token is present in the document, its value is 1, otherwise 0. For the TF-IDF weighting method, it is the product of two statistics, the term frequency and inverse document frequency. The consideration of the term frequency assumes that the more frequent a token appears in the document, the more important the token for the topics of the document. At the same time, the inverse document frequency is used to offset the impact of common words shared by all documents. But the BOW method suffers from the following issues:

Sparsity, most of the words in the vocabulary are absent from individual documents, resulting in the term-document matrix consisting of a lot of unwanted zeros. This challenges the machine learning algorithms, easily causing overfitting problem for the models since a lot of instances, relationships between features, are not available in the training data.

The document representation (a TF-IDF or a binary vector) does not take the word order into account and only considers the occurrence of a word independent of the others, which is not true from both semantic and syntactic point of view.

High dimensionality. Large corpora can have at least thousands of words (features). On top of this, if 2-grams or 3-grams are included, the number of features per document increases significantly. It could generate an even sparser term-document matrix and lead to insufficient RAM problem when we try to hold the entire matrix in the RAM. Not all features are important, and modelling the data with so many features is easy to lead to overfitting for supervised learning when no sufficient labelled samples are provided.

### 4.2 Word2Vec

The Word2vec algorithm consists of a bunch of related models that are used to generate a distributed representation of word embeddings. These models are the continuous bag-of-words (CBOW) and the skip-gram as illustrated in Figure 1, in which the left scheme is the CBOW model, and the right side is the Skip-gram model.

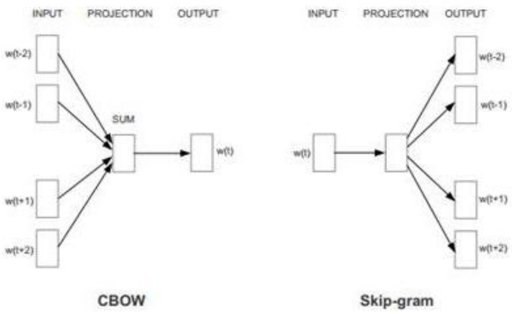

**Fig. 1.** The illustration of CBOW and Skip-gram models in Word2Vec with courtesy of T. Mikolov etc. [2].

In the CBOW architecture, the model predicts the current word from its surrounding context words within a context window centred at the current word w(t), while for the skip-gram model, it predicts its surrounding context words within a window according to the current word. A Word2vec model can be trained with the hierarchical Softmax and/or negative sampling method. The hierarchical Softmax method uses a Huffman tree to reduce the computational complexity while the negative sampling method accomplishes this goal and improves the vector quality of low-frequency words by only sampling a few negative samples for updating the vectors for each iteration instead of updating all negative words' vectors. Also, the high-frequency words are down-sampled and the low-frequency words are up-sampled by lifting their frequencies in the negative sampling process. These models are the two-layer shallow neural networks. Word2vec takes as its inputs the high dimensional one-hot vectors of a large corpus of texts and produces a vector space of several hundred dimensions such that each unique word in the corpus is represented by a dense vector in the embedded low-dimensional vector space. A very salient feature of this kind of word embeddings is that word vectors are such points in the vector space that semantically similar words' vectors are close to each other. This offers the great benefit that for a word, we can infer its semantically similar words from the vector space if the word's vector is known and hence word2Vec has attracted tremendous attention in text analysis. One way of using the word vectors for document classification and clustering analysis is to take the averaged word vectors in a document as shown in Figure 2, forming a single vector to represent a document. Another way of using word embeddings for text classification is for sentence classification with CNN in which the individual words' vectors of the sentence are taken as the inputs for training the classification model [4].

### 4.3 Doc2Vec

Doc2Vec [3] is an extension of Word2Vec that tries to model a single document or paragraph as a unique real-valued dense vector in addition to generating the word vectors for individual words in the corpus. Just like word vectors generated by word2Vec, which provide semantic inference for individual words, a document vector generated by doc2Vec, the Paragraph id shown in Figure 2, can be thought of reflecting some semantic and topic information of the document. As a result, the document vectors of similar documents tend to be close to each other in the vector space. It is very useful for document classification or clustering analysis with the generated single document vector. Applications include sentiment analysis [5] and text classifications.

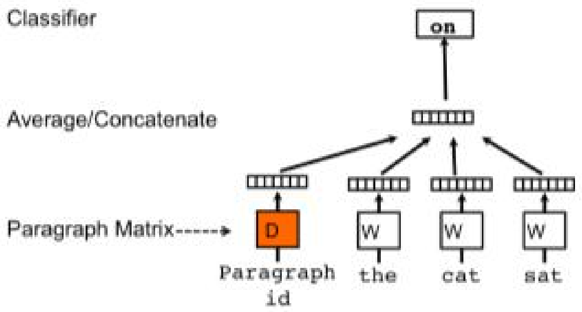

**Fig. 2.** A framework for learning paragraph vector with courtesy of Quoc Le, etc. [3].

## 5 The unsupervised clustering algorithm

To find effective medical treatment solutions from the EHRs for any input patient's symptom description, we propose to first locate the similar symptom descriptions in the EHRs and then find their corresponding medical treat solutions, medication prescription, and doctors' contact information for consultation purposes. So, the similar symptom descriptions act as the indices to the relevant and effective medical treatment solutions. Having generated a vector for each document/symptom description, it is natural for us to cluster all similar documents with respect to the input document. For this, we measure the similarity between any two documents by calculating the numerical value of similarity metric between the two documents' vectors. The higher the similarity, the more similar they are and the more relevant the corresponding medical treatment solution. We calculate the normalized cosine similarity between any two vectors A and B as the quantitative metric to measure their similarity:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|_2\|\mathbf{B}\|_2} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}} \qquad (1)$$

For each patient's document of symptom description, we select the top 10 treatment solutions according to the sorted similarities of their symptom descriptions in the EHRs in decreasing order. Furthermore, to make sure that all top 10 medical solutions are relevant, an additional condition must be satisfied that their corresponding similarities are higher than a threshold, which is learned with some labelled records.

## 6 Experimental results and analysis

We have first investigated 4 document representations: TF-IDF vector for tokens, binary vector for tokens, the averaged word embeddings with word2Vec, and doc2Vec. For optimizing the word2Vec, we have investigated 4 models: CBOW with hierarchical Softmax, CBOW with negative sampling, skip-gram with hierarchical sampling, and skip-gram with negative sampling. Finally, the CBOW model with negative sampling is selected with the optimized hyperparameters: window=3, dimension=200, the threshold for low frequency threshold is 1e-5， 5 samples are used for negative sampling, and iterNum = 10. However, from experimental results, we find out that they do not do well for our specific data set. The returned results are not all relevant to the input document by our visual checking. After careful study with additional investigations, we find out that the binary vector of keywords extracted by Han LP [1] works best by visually checking the returned top-10 similar documents for individual input documents. The medical documents are comparatively short and there are some discrepancies for the words used by different doctors and hospitals to describe the similar symptoms, as a result, the first two BOW vector representations generate very high dimensional and very sparse vectors for documents, and even for similar diseases, the vectors representations do not reflect much similarity, so the clustering performance is not very satisfactory. The word embedding method of word2Vec could provide semantic information for words, but the averaged word vector introduces a lot of noise to describe the document. So, in the recommended results obtained by the averaged word2Vec embeddings, some of them are very different from the input document. The doc2Vec could theoretically provide semantic information for each document reflected in the generated vector, but for our short EHR data, the added document tag could not provide much semantic information about the documents, so the vector quality is not high and most of the recommended results are not relevant to the input document after visual check. On the other hand, the use of the keywords for binary vector representation greatly reduces the dimensionality of the binary vector and at the same time offers semantic information for symptom descriptions. So, it accomplishes much better recommendation results in our medical POC project. All recommendation results are evaluated by visual checking. The system is implemented in Java and has been successfully tested by the testing team.

## 7 Conclusion and future work

In this paper, we develop NLP and unsupervised machine learning algorithms to leverage the insights from EHRs for recommending medical treatment solutions for patients in the countryside and small cities when they have schizophrenia and the local hospital doctors do not have expertise to tackle such challenges. The binary keyword vector representation is very efficient for representing patients' symptom descriptions in the EHRs for finding the similar disease symptoms. In the future, we will develop other semantic vector representations for document clustering, try other clustering algorithms, and develop quantitative metrics to measure the quality of the recommendation results for comparing different algorithms.

## References

1. https://datascience.shanghai.nyu.edu/hanlp
2. Mikolov T, Chen K, Corrado G, Dean J, (2013), "Efficient Estimation of Word Representations in Vector Space", arXiv:1301.3781, Computation and Language.
3. Le Q, Mikolov T, (2014), "Distributed Representations of Sentences and Documents", International conference on machine learning.
4. Kim Y, (2014), "Convolutional Neural Networks for Sentence Classification", Proceedings of the Conference on Empirical Methods in Natural Language Processing.
5. https://www.datasciencecentral.com/profiles/blogs/sentiment-analysis-of-movie-reviews-2-doc2vec